UUCS-86-006
Rev. 1

TILER USER'S GUIDE

Steven R. Jacobs

Kent F. Smith

January 12, 1987

VLSI RESEARCH GROUP
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF UTAH
SALT LAKE CITY, UTAH 84112
(801) 581-8580

Copyright © 1987 University of Utah
All Rights Reserved

This work was supported in part by *Defense Research Projects Agency* under Contract Number DAAK1184K0017. All opinions, findings, conclusions or recommendations expressed in this document are those of the author(s) and do not necessarily reflect the view of DARPA.

CONTENTS

LIST OF FIGURES

CHAPTER 1

TILER BASICS

Tiler is a special interactive editor used for designing VLSI circuits using the Path Programmable Logic (PPL) methodology. PPL cells are inserted into a rectangular grid by typing characters that represent the cells. Each available cell will have a character that represents it, but some characters might not have a cell "defined". When a character is typed that has no PPL cell defined for it, an appropriate error message will be displayed. All ordinary printable characters are assumed to be associated with PPL cells. The exact cell that is associated with each character will depend on which TECHNOLOGY has been specified. When **tiler** is first started, the user is prompted for the technology to be used for designing a PPL circuit. When a technology name is given by the user, a technology file is loaded into **tiler**. This file contains information about the process used to define the PPL cells (NMOS, CMOS, GaAs, etc.), the number of row and column wires in each PPL cell, the spacing of cells on the PPL grid, and information about each of the cells that are defined for the given technology.

1.1 TILER Command Types

Tiler is modeled after the EMACS text editor, and has many similarities to EMACS. However, **tiler** is designed for editing large two-dimensional arrays of characters, so there are some important differences from EMACS. Most **tiler** commands are executed by typing control characters. Control characters are usually non-printing characters, and are displayed with '^' preceding the character, as in ^A, ^B, ^C, etc. Most of these commands take effect immediately, and the effect of the commands can be seen on the screen after

the command has been executed. Some control characters act as prefix characters, and must be followed by additional characters in order to form a complete command. The prefix characters used by **tiler** are: <esc>, ^X, and ^Z. The character ^U may be used for specifying arguments to commands, but when ^U is followed immediately by a command character, it may be thought of as a prefix character (arguments are discussed further in another paragraph). **NOTE:** Typing ^[is the same as typing escape or altmode.

When a prefix character is typed, it is echoed at the bottom of the screen below the **tiler** status line, and the cursor remains positioned next to the prefix, waiting for the remaining characters of the command. Commands which use <esc> as a prefix are called META commands, and the <esc> character is echoed as "M-". There are two varieties of META commands. An 'ordinary' META command is formed by typing the <esc> key followed by any ordinary printable letter, number, or other printable character. A META-CONTROL command is formed by typing the <esc> key followed by a CONTROL character. META-CONTROL commands can also be executed by typing a ^Z followed by either an ordinary character or a control character. **NOTE** to EMACS wizards: **tiler** also supports the use of a separate META key. On terminals that support a META key, META commands may be entered by holding down the META key and typing a regular character or a control character.

The ^Z prefix is mainly included as a shortcut for entering META-CONTROL commands, and is echoed as "M-C-" or as "^Z" whenever it is typed. When ^Z is followed by a character in the set {'A'...'Z', 'a'...'z', '[', '/', ']', '_', '^', '@'}, the character is converted to upper case and treated as if the control key had been held down while the character was typed. For example, the key sequences '<esc><ctrl-a>', '<esc><ctrl-A>', '<ctrl-Z><ctrl-a>', '<ctrl-Z><ctrl-A>', '<ctrl-Z>a', and '<ctrl-Z>A', will all execute the same META-CONTROL command.

The ^X prefix is used for eXtended commands or eXtreme commands, and is echoed as ^X whenever it is typed. Many of these commands are used

for reading or writing files, or other operations which require additional information from the user. Appropriate prompts are used when executing these commands. Some of the ^X commands are used to repeat a command "as much as possible". Other ^X commands are placed in the ^X category simply for compatibility with EMACS or because there was no other preferred place to put them.

Lower case letters and upper case letters are treated as the same letter when used in commands, so that ^a is the same as ^A, and M-a is the same as M-A. However, when characters are typed to indicate that a CELL should be inserted into the PPL array, the case of the letter is important, so that 'a' and 'A' may be entirely different cells. When cells exist for both the upper and lower case of a particular character, that character must be typed with the correct case in order for **tiler** to insert the correct cell. If a cell exists for only the upper case letter or only the lower case letter (but not both), then **tiler** will insert the correct cell when typed in either upper case or lower case.

1.2 Aborting TILER commands

If a prefix character is typed by mistake, or if it becomes necessary to not execute a partially typed command, the entire command can be aborted at any time by typing ^G. This will ring the bell on the terminal and return the cursor to its correct position in the PPL array. However, once the execution of a command has begun (usually by typing the last character of the command), nothing can be done to interrupt the execution of the command, and any characters typed while a command is being executed will be treated as commands to be executed after the current command has been completed. Several commands may be typed "ahead", and will be executed one after the other just as they would be if the user had waited for each command to be completed before typing the next command. When several commands are typed rapidly, **tiler** will try to complete the execution of all commands before updating the appearance of the screen. Also, commands that are typed while the screen is being changed will be executed before the screen update is completed.

1.3 Passing ARGUMENTS to Commands

Integer arguments may be passed to commands by typing either <esc> or ^U, followed first by the digits of the argument, and then by the characters which form the command. Arguments may be either positive or negative, but some commands may act strangely with negative arguments. Errors made while entering arguments may be deleted using the <del> or <backspace> keys (or by ^H, which is the same as <backspace>). Arguments are usually interpreted as the number of times to repeat a given command. When no argument is passed to a command, a default value of *one* is used. If ^U is followed immediately by a command character, an argument of *four* is used for the command. Arguments that are a power of four can be specified by typing ^U more than once. For example, ^U^U specifies an argument of 16, and ^U^U^U specifies an argument of 64. This feature can be useful for moving the cursor over a large distance without having to type in an exact argument. Note, however, that digit keys typed immediately after one or more ^U characters will cause the digits to specify the argument, just as if only one ^U had been typed, or as if <esc> had been typed before the digit keys. Note also that this feature works only with the ^U character, and NOT with the <esc> character. Two <esc> characters typed in sequence form a M-<esc> command, which has nothing to do with specifying an argument.

The usual method of passing an argument to a command is to type an <esc> character (which will be echoed as "M-") and some digit keys ("0".."9", "-") before typing the command keys. If the <esc> key is followed by a non-digit key, the <esc> acts as the META prefix to the character, and the usual default argument of *one* is used for the command. In order to pass an argument to a META command, first enter the argument by typing either <esc> or ^U followed by the appropriate digits, then type <esc> as the META prefix, then type the desired command character.

When a single argument is followed by a cell character, that argument is used as the modifier value for the cell. If two arguments are used with a cell

character, the first argument typed is used as a repeat count and the second argument is used as a modifier value. Note that to use repeat counts with unmodified cells, the second argument must be zero. When repeat counts are used, several cells are inserted in adjacent locations in the PPL array. This works fine for all cell characters except those in the set {'0'...'9', '-'}. To specify a repeat count for these cells, type in the repeat count (and the modifier value) in the usual fashion, then type ^Q (hold down the CONTROL key and hit the 'Q' key), then type the cell character to be inserted. This method works for any cell character (or for **tiler** commands, for that matter), but is only necessary when the cell character is a digit or a '-'. The ^Q method is also needed for applying arguments to the <del> and ^H (backspace) commands.

NOTE: The '+' character cannot be used to start an argument, since it will be interpreted as either the '+' cell (when ^U starts the argument) or as the M-+ command (when <esc> starts the argument).

1.4 Getting HELP from TILER

The main HELP utility of **tiler** can be accessed by the ^X-? command, or by <ctrl-underscore> as in EMACS. Since the exact key needed to type the <ctrl-underscore> character varies from terminal to terminal, the <ctrl-X>? command is the preferred method for getting help from **tiler**. This will display the prompt "Doc (? for help):" and wait for a character to be typed to indicate the help category. Typing a '?' in response to this prompt will produce a listing of the available help categories. Listings of all commands or all cells may be obtained by using the main help facility, as well as a description of some advanced features of **tiler**. First time users of **tiler** should take the time to read all of the help categories that are available.

The commands M-? and M-/ may be used to get a description of a command character or a PPL cell. The user is prompted to type in the command character or the cell character. Descriptions of commands and cells will be displayed one screenfull at a time. When descriptions require more than

one screen, the 'y' key or the <space> key may be used to move to the next screen, typing any other key will 'quit' the help command without displaying the rest of the description.

The commands ^Z-? and ^Z-/ may be used to display the message associated with the most recent error encountered by **tiler**. **Tiler** will usually print error messages as errors occur, but minor errors will only ring the bell at the terminal without displaying a message. ^Z-? and ^Z-/ allows these messages to be examined.

CHAPTER 2

BREAKS BETWEEN PPL CELLS

Two characters are used to represent a single PPL grid location: one character for the cell itself, and one character for the breaks which occur to the *LEFT* and/or *BELOW* the cell. At each PPL grid location, the break character is placed to the left of the cell character. The exact display representation of breaks between PPL cells will depend on the type of terminal being used. Terminals that support character underlining will use underlines, placed below the break and cell characters, to represent breaks in column wires. Terminals that do not support character underlining must use different break characters to represent the various combinations of row and column breaks.

The break characters are used only for displaying breaks in the "primary" row and column wires. The primary row wire is usually named ROW, and the primary column wires are usually named RCOL (for Right COLumn) and LCOL (for Left COLumn). On terminals that are able to underline characters, the only break character used is the '|' character. This character is used to indicate a break in the ROW wire. Breaks in the LCOL wire are displayed as underlines below the break character, and breaks in the RCOL wire are displayed as underlines below the cell character. Thus underlining terminals are able to display breaks directly as lines that "block" the signals from passing between neighboring cells.

On terminals that cannot underline, a different break character is used for each distinct combination of row and column breaks. This requires more effort on the part of the user to interpret the meaning of the break characters, so it is recommended that terminals with underlining capability be used whenever possible. Although strange breaks characters must be used to indicate breaks

in many instances, **tiler** will attempt to use '|' for row breaks and underlines for column breaks whenever possible. Figure 1 lists the break charaters used by **tiler**. These same characters may be typed on the terminal to break the corresponding row and column wires. Breaks are additive, so typing a break character will add new breaks to the current cursor position but will not remove any breaks that are already there. The <backspace> or <del> keys may be used to remove all breaks from the current cursor position. The '|' and '_' keys may also be used for entering arbitrary combinations of row and column breaks. Some cell sets may use other keys, in addition to those listed in Figure 1, for entering breaks.

BREAK	meaning
	no primary wires broken
|	ROW wire broken
,	LCOL wire broken
.	RCOL wire broken
=	LCOL and RCOL broken
!	ROW and LCOL broken
%	ROW and RCOL broken
&	all primary wires broken

Figure 1: BREAK characters on terminals without underline capability

Some cell sets will have secondary row wires and/or secondary column wires. Breaks in these wires are not displayed in the main display area of the screen, but they are displayed in the **tiler** status line. Whenever the cursor is directly over a cell that has broken secondary wires, the status line will include an indication of all wires that are broken. Thus when "invisible" wires are broken *AT THE LOCATION OF THE CURSOR*, a display of the form "B:_|._||" will appear in the status line. This particular string would appear for a cell set that has one secondary row wire and one secondary column wire. This particular example shows that both secondary wires are broken, but none of the primary

wires are broken. The "B:" part indicates Breaks, and each broken wire is indicated by an underline, while each "connected" wire is displayed as a vertical bar. The '.' can be thought of as a hinge that separates the *row* wires on the left side of the hinge from the *column* wires on the right side of the hinge. Using this hinge to pivot the row wires by 90 degrees in the clockwise direction, the vertical bars can be visualized as connected horizontal row wires. The character that represents the main ROW wire is always immediately left of the hinge, and all secondary row wires (if any exist) are represented by characters that appear further to the left. The rightmost character is the RIGHT column wire, and the character just left of the RIGHT column wire is the LEFT column wire. If any secondary column wires exist, they will be displayed between the LEFT column wire and the hinge. Admittedly, this is a bit clumsy, but it is less confusing than attempting to display all wire breaks in the main display area.

The '|' and '_' break characters may be used with modifier values to enter any combination of row and column breaks. The '|' character is used for row breaks, and the '_' (underline) character is used for column breaks. When a modifier value is passed as an argument to one of these break keys, the modifier value is used to indicate which wires are to be broken. Modifers may be passed to these breaks (and to PPL cells) by typing the <esc> key, followed by one ore more digit keys, before typing the cell character. Figure 2 shows which wires are broken by combining different modifier values with the '|' and '_' keys. The examples assume a cell set with two row wires and three column wires. Entries marked as "(not shown)" represent combinations of breaks that do not generate a B:||.||| type display in the status line, since no "invisible" wires are broken. Note that the pattern of broken wires can be thought of as a binary number that matches the modifier value used. If no modifier value is used, or a value of zero is used as a modifier value, a value of one is used as a default value.

Modifier value	Break cell	Break pattern (as displayed)	Notes
3	'|'	B:__.|||	
2	'|'	B:_|.|||	
1 (0)	'|'	B:|_.|||	(not shown) (default)
7	'_'	B:||.___	
6	'_'	B:||.__|	
5	'_'	B:||._|_	
4	'_'	B:||._||	
3	'_'	B:||.|__	(not shown)
2	'_'	B:||.|_|	(not shown)
1 (0)	'_'	B:||.||_	(not shown) (default)

Figure 2: *Using modifiers with the '\|' and '_' Break cells*

CHAPTER 3

KEYBOARD MACRO DEFINITIONS

Tiler has the capability of "remembering" a command sequence typed by the user, and executing the command sequence just as if it were a built-in **tiler** command. Such a sequence of commands is called a keyboard macro. A keyboard macro may consist of almost any sequence of **tiler** commands and/or cell characters. If you find that you are about to execute the same sequence of commands 100 times, then you can define a keyboard macro which will execute the command sequence once, and then pass an argument of 100 to the keyboard macro to repeat the entire command sequence 100 times.

The command ^X-(is used to start defining a keyboard macro. After the ^X-(command is given, **tiler** will save each command and/or cell character typed by the user. The commands are also executed as they are typed, so that the user can see the effect of each of the commands in the keyboard macro. Thus the process of creating a keyboard macro also "executes" the keyboard macro for the first time. When a keyboard macro is being defined, the string "Def" is added to the **tiler** status line.

The command ^X-) is used to complete the definition of a keyboard macro. After the ^X-) command has been given, the keyboard macro can be executed by the ^X-E (Execute macro) command. Arguments preceding the ^X-E command are treated as the number of times to repeat the execution of the keyboard macro, but an argument that is less than or equal to zero will prevent the macro from being executed. If no argument is specified, a default value of one is used. Any error condition encountered during execution of a keyboard macro will cause the execution of the macro to terminate.

The only command that is not allowed in a keyboard macro is the ^X-E command. **tiler** will not attempt to execute a keyboard macro before it is completely defined.

NOTE: Whenever the ^X-(command (Define keyboard macro) is given, a new macro definition is started, and any keyboard macro defined previously will be lost. If the ^X-(command is given while a keyboard macro is partially defined, **tiler** will throw away the keystrokes that were saved since the first ^X-(command was given, and begin the definition of the keyboard macro with the next command. However, keyboard macros may be "assigned" to keys on a "USER" keyboard, so that many frequently used macros can be saved and executed quickly.

3.1 User Defined Commands

Tiler commands may be divided into two classes. The first class of commands are the "built in" **tiler** commands, which are usually executed by either one to two keystrokes (one keystroke for cells and CONTROL commands, two keystrokes for META commands, META-CONTROL commands, and ^X commands). The second class of commands are the User Defined (USER) commands. The USER commands may be thought of as commands which are typed on a "USER keyboard". The <esc>-U command may be used to change back and forth between the USER keyboard and the standard **tiler** keyboard. The string "USER" appears in the **tiler** status line whenever USER mode is in effect. When a user-defined command is typed from USER mode, **tiler** executes the keyboard macro that has been assigned to that command. If a command is typed (while in USER mode) that has not been defined by the user, the built-in **tiler** command is executed. Any error condition encountered during the execution of a USER command will cause the execution of the command to terminate.

USER commands are actually keyboard macros that are "assigned" to keys on the USER keyboard. The ^X-D (Define key) command is used to "assign"

the current keyboard macro to a key on the USER keyboard. Once a keyboard macro has been defined by using the ^X-(and ^X-) commands, and assigned to a key by the ^X-D command, the USER command may be executed either by the ^X-E command, or by typing the defined key while in USER mode. After a USER key is defined, the keyboard macro may be changed without changing the definition of the USER key. Thus several different USER keys may be defined at any one time. A USER key may be redefined at any time by using the ^X-D command again. The ^X-U command can be used to "undefine" a USER key. The ^X-K command may be used to display the definition of any defined USER key. Key definitions displayed by the ^X-K command are displayed as a single string of keystrokes, with '^' used to indicated control characters and <esc> used to indicate META commands. Key definitions are not always displayed in the exact form that they were typed, since they are converted into a uniform internal format when they are created.

Sometimes it is desirable to change keyboards only for the duration of a single command. When this is necessary, it is inconvenient to type <esc>-U to switch keyboards, then type the command, then type <esc>-U again to switch back to the original keyboard. **Tiler** provides a special command prefix in order to avoid this inconvenience. The command prefix <esc><esc> may be used to switch to or from the USER keyboard for the duration of a single command. When this prefix is typed, the string "<*>" appears on the **tiler** command line to indicate that the "other" keyboard has been activated.

IMPORTANT NOTE: If a keyboard macro has been defined for the <esc>-U command and USER mode is in effect, the only way to switch keyboards is to type the command sequence <esc><esc><esc>U. The first two escapes tell **tiler** to switch to the standard keyboard for the <esc>-U (META U) command, which will then execute the "Toggle USER mode" command.

Some keys cannot be user-defined. Prefix keys such as <esc>, ^X, ^Z, and ^U cannot be redefined. Some commands (such as ^G, <esc>^G, ^X-^G, and ^Z-G) probably shouldn't be defined, in order to avoid confusion.

USER cells may also be defined by assigning a keyboard macro to keys which type ordinary printing characters. These keys should probably be reserved for commands which insert cells or groups of cells, but **tiler** places no restrictions on what type of keyboard macros are assigned to keys.

USER keys may also be placed in keyboard macros. This allows the execution of USER keys to be nested, calling other USER key definitions. This feature must be used with care to obtain correct results. It is even possible to create a USER key definition which calls itself. Any error condition encountered during execution of a USER command will terminate execution of ALL LEVELS of nested commands. The nesting of USER keys is arbitrarily limited to 128 levels. This is done to avoid having **tiler** attempt to execute an infinite sequence of USER commands which are nested improperly, and to avoid having **tiler** crash due to a subroutine stack overflow, which would lose all previous work.

Tiler can also load USER key definitions from an external file by using the ^X-X command. The ^X-X (eXecute command file) prompts for that name of a file containing **tiler** commands to be executed. Commands which define keyboard macros and assign them to USER keys may be placed in a separate file and loaded whenever they are needed.

CHAPTER 4

TILER COMMAND DESCRIPTIONS

4.1 Control character commands:

NOTE: if the commands ^F, ^B, ^N, ^P are 'assigned' to cursor keys on the terminal, then the keys <esc>, ^U, and ^X can be used as prefixes to the cursor keys to control the amount of movement caused by these commands.

^@ (NULL). Set MARK. Argument between 0 and 9. (Default 1) Saves the current cursor location for future reference.

NOTE: MARK 1 is the "default" mark, which may be changed by some of the window commands. For remembering "special" PPL coordinates for long periods of time, use any of the other MARKS. Use ^I or TAB to move to marks.

^A move to first column on screen. ^X^A moves to far left edge of PPL, changing screens if necessary.

^B move Backward (left) one or more columns. May be preceded by an argument. M-^B moves backward by one or more screens.

^C Suspend **tiler**. **Tiler** may be re-entered without losing anything.

^D Delete PPL cell (and surrounding breaks) at current cursor location. With argument, deletes many cells, moving in the default direction. To delete a cell without deleting surrounding breaks, use M-^D.

^E move to End of PPL (but remain on this screen). Moves past the last PPL cell or break character of the row, without leaving the screen. ^X^E moves past the last PPL cell or break character of the entire PPL row, changing screens if necessary.

^F move Forward (right) one or more columns. May be preceded by an argument. M-^F moves forward by one or more screens.

^G Universal abort command. Type this at any time to abort a partially typed command.

^H (BACKSPACE). Delete all break information at the current cursor

location. With argument, deletes specified number of breaks, moving in the default directions. Some break information is "built-in" to cells and cannot be deleted. This command is also executed by the RUBOUT key.

^I (Tab). Move to PPL grid location, given as two arguments. The cursor is moved to the row and column specified by the two arguments. If less than two arguments are given, the cursor is moved to a MARK which was set by a M-^I command or a ^@ command. A single argument between 0 and 9 may be used to move to any of the 10 available MARK positions.

^J No op. This command is always ignored by **tiler**, but is different from "undefined" commands because ^J will not generate an "undefined command" error message.

^K Kill WINDOW. Argument between 0 and 9 (default 1). Delete the cells from the specified window, saving them on the kill ring. Blocks of cells which are on the kill ring may be retrieved with the ^Y (yank) and M-Y (rotate kill ring and yank again) commands. The ^K command requires the cursor to be at the window origin in order to execute the command, unless the ^K is preceded by a ^Q.

NOTE: This version deletes the window AND breaks surrounding the window. To delete the cells and breaks inside the window without deleting breaks on the edges of the window, use M-^K. Blocks of PPL cells may also be copied from one buffer to another by using the kill ring.

^L Repaint the screen.

^N move down to Next row. May be preceded by an argument. M-^N moves down one or more screens. M-> moves to lower edge of PPL, changing screens if necessary.

^P move up to Previous row. May be preceded by an argument. M-^P moves up one or more screens. M-< moves to top edge of PPL, changing screens if necessary.

^R Reverse search for a specified cell. The user is prompted for a cell to search for. Reverse searches start from the current cursor position and move backward in the direction opposite of the default direction. Reverse searches will move toward the left side of the PPL array unless the default direction has been changed by a M-D command. If a reverse search reaches the edge of the PPL array without finding the cell, searching is continued on the previous (lower numbered) row or column. A second ^R may be typed instead of a cell character to search for another occurance of the last cell searched for. A ^S may

be typed to reverse the direction of the search. Any other command character typed will cause the search to end so that the command can be executed. ^G may be used to quit searching. Arguments may be used to search past several occurances of a cell. For example, an argument of 10 would cause a search for the tenth occurance of the cell. If the cell is found fewer times than specified by the argument, the cursor is placed at the last occurance that was found.

NOTE: Reverse searches halt at the first column or row of the the PPL array, and do not wrap around to the other side of the array. Thus a reverse search may fail even though the cell exists in the PPL array. If necessary, a ^S may be typed to reverse the direction of the search, in order to verify that the cell does not occur elsewhere in the array.

^S Search for a specified cell. The user is prompted for a cell to search for. Searches start from the current cursor position and move forward in the default direction. The default direction will be toward the right side of the PPL array unless it is changed by the M-D command. If a search reaches the edge of the PPL array without finding the cell, searching is continued on the next (higher numbered) row or column. A second ^S may be typed instead of a cell character to search for another occurance of the cell specified in a previous search command. A ^R may be typed to reverse the direction of the search. Any other command typed during a search will cause the search to end so that the command can be executed. ^G may be used to quit searching. Arguments may be used to search past several occurances of a cell. For example, an argument of ten would cause a search for the tenth occurance of the cell. If the cell is found fewer times than the number specified by the argument, the cursor is placed at the last occurance that was found.

NOTE: Forward searches halt at the last column or row of the the PPL array, and do not wrap around to the other side of the array. Thus a forward search may fail even though the cell exists in the PPL array. If necessary, a ^R may be typed to reverse the direction of the search, in order to verify that the cell does not occur elsewhere in the array.

^Q Quote. Some error conditions in **tiler** may be "bypassed" by preceding a command with ^Q. Severe error conditions cannot be avoided. If you think that **tiler** is restricting you from doing something that you really want to do anyway (even though it is probably a mistake), try preceding the command by ^Q. **Tiler** may or may not let you get away with it. If what you want to do really is a mistake, it WILL catch up with you sooner or later, so don't use this feature unless you really know what you are doing. You may choose to always ignore these

"warnings" using the M-Q command. Do so at your own risk ("Danger!" will appear in the **tiler** status line whenever this mode is in effect).

Examples of things you can get away with by using ^Q are:

* Inserting cells at improper locations (wrong row/column, etc.).
* Deleting windows when not at the proper origin.
* Any change that leaves a cell at an improper location. (Inserting or deleting PPL rows/columns, moving windows, copying windows, inserting files, etc.)

^U Universal argument. When followed by '-' and/or digit keys, and then a command key, the specified number is passed to the command as an argument. All commands may be passed arguments, but some commands will ignore any arguments that are passed to them. If ^U is followed immediately by a command character, a value of FOUR is used for an argument. Several ^U's may be used to specify larger arguments, with each ^U multiplying the argument value by four (thus ^U^U = 16, ^U^U^U = 64, etc). This feature is useful for specifying a "small", "medium", or "large" argument. For example, ^U^U^F will move forward by 16 columns (about half a screen), and is much faster than typing ^F sixteen times.

^V move down one or more screens. May be preceded by an argument.

^W Copy WINDOW. Copies a window onto the kill ring without deleting cells from the PPL array. This has the same effect as ^K followed immediately by ^Y.

^X command prefix for eXtended commands.

^Y Yank. Copies a block of cells from the kill ring to the current cursor location. By using the ^K and ^Y commands, large blocks of cells can be either moved (^K, move to new location, ^Y) or copied (^K then ^Y, move to new location, ^Y again). When this command is executed, the default window is set to surround the yanked window, so that a ^Y followed immediately by a ^K will 'undo' the ^Y command. As in EMACS, M-Y may be used to rotate through the kill ring to find blocks of cells from several previous ^K commands. Blocks of PPL cells may also be copied from one buffer to another by using the kill ring.

^Z command prefix for META-CONTROL commands.

^[(same as <esc>). Command prefix for META commands.

4.2 META commands:

M-A move to start of PPL (but remain on this screen). Moves to the first PPL cell or break character of the row, without leaving the screen. If the row has no cell or break character on the screen, moves to the left edge of the screen. ^X-A moves to the first PPL cell or break character of the entire PPL row, changing screens if necessary.

M-C incremental CIF check/correct. Moves the cursor to the location of the next CIF error and displays a message indicating the nature of the problem. If an argument is passed to this command, **tiler** will attempt to correct the error rather than report an error message. An error will usually be a cell type mismatch, meaning that two incompatible cells have been placed next to each other. The search proceeds from left to right in columns, and from bottom to top in rows, starting from the current cursor location. If no errors are found, the cursor is left at its original location and a message is displayed indicating that no new errors were located. The number of errors that were "repaired" is also reported. This command is useful for locating and correcting design rule violations, one at a time. Start at the lower-left corner of the PPL array (row 0, column 0) to find the first CIF error. The ^X-C command may be used to create a file which describes all existing design rule violations.

M-D Set default Direction. Prompts for the direction to move after inserting cells. Respond with ^F, ^B, ^N, or ^P.

M-F Fill window with cells. **Tiler** prompts for the cell character. This may be used to fill an entire region with cells. **Tiler** tries to fill in all uncovered areas of the specified window with the given cell.

M-L Set number of Rows of PPL to be displayed on the screen. Limited to the maximum screen width of the terminal, but may be less if desired.

M-M Modify cell. Set the modifier value of the cell to the given argument. Cell modifiers are used to allow minor variations in the characteristics of some cells. In NMOS cell sets, modifiers might be used to add pullups to certain signal wires within a cell, or to vary the strength of internal pullups.

M-O Move to window. Same as M-[.

M-Q Toggle QUOTE mode. QUOTE mode prevents **tiler** from generating error messages when certain placement restrictions are violated. This should only be used by experienced users under special conditions. When QUOTE mode is in effect, "Danger!" will appear in the **tiler** status line. To get out of QUOTE mode, execute M-Q a second time.

M-R Toggle RIGID mode. RIGID mode is used to prevent **tiler** commands from changing the size of the PPL array. When RIGID mode is not in effect, some commands will automatically expand the PPL array if the expansion is necessary in order to execute the command.

NOTE: When RIGID mode is in effect, the commands ^X^I (insert PPL rows/columns) and ^X^K (delete PPL rows/columns) will generate an error message to indicate that size of the PPL array cannot be changed.

M-T Terminal display mode. Argument is used to determine type of characters use for displaying breaks, as described in the following table:

arg	display mode
0	use for breaks, NO underlining
1	(default) use underlining when possible
> 1	use underlining all column breaks

NOTE: Attempting to use underling capabilities that do not exist may result in NO display of break information.

M-U Toggle USER mode. In USER mode, the user defined keyboard becomes the primary keyboard. Defined keys then take priority over the standard keys, so that when a defined key is pressed, the keyboard macro associated with that key is executed. Command keys which have not been user-defined will still execute the standard **tiler** command for that key. When in USER mode, a standard **tiler** command may be executed by typing the <esc> key *twice* before typing the command. The word "User" appears in the **tiler** status line when USER mode is in effect. When USER mode is NOT in effect, a command which has been user-defined may be executed by typing the <esc> key *twice* before typing the command.

M-V move up one or more screens.

M-W Set screen Width. Sets the number of columns of PPL to be displayed on the screen. Limited to the maximum screen width of the terminal, but may be less if desired.

M-Y Yank again. This command may be used only after a ^Y or another M-Y. Following a ^Y command, M-Y may be used to rotate through the kill ring to find blocks of cells from several previous ^K commands. If M-Y is attempted when a block is too large to yank or at an improper row/column location, the kill ring is still rotated so that another M-Y may be used to continue rotating through the kill ring.

M-Z Toggle ZOOM mode. (Not yet implemented)

M-< move to top edge of PPL array, remaining in the same column.

M-> move to bottom edge of PPL array, remaining in the same column.

M-[Move to WINDOW. Argument between 0 and 9. (Default 1). Places the cursor at the origin (lower-left corner) of the window, and sets the default MARK at the upper-right corner of the window. These two corners of the window can then be easily located by using the "exchange point and mark" command ^X^X to move the cursor back and forth between the lower-left and upper-right corners of the window. The upper-right corner of the window may also be located using the M-] command.

M-] Move to WINDOW. Like M-], but moves to upper-right corner of the window.

M-{ Set lower-left corner of WINDOW. Argument between 0 and 9 (Default 1).

M-} Set upper-right corner of WINDOW. Argument between 0 and 9 (Default 1).

M-~ Tell **tiler** to pretend that the current file has not been modified.

M-;,
M-: Toggle COORDINATE ECHO mode. Turns on/off the Rxx:Cxx in the status line. Turning off the row:column indicator speeds up execution of commands, because updating the Rxx:Cxx indicator can require more time than the the execution of the command itself. Especially speeds up ^F, ^B, ^N, ^P.

M-/,
M-? Describe a **tiler** command or PPL cell. You type the command characters. Displays the 'long' description of the given command.

4.3 META-CONTROL commands:

NOTE: The META-CONTROL command prefix may be typed either ^Z or as <esc> followed by control character. On terminals that have a META key, META-CONTROL commands may be typed by holding down both the META and CONTROL keys while typing the command character.

^Z-B move Backward (left) one or more screens. May be preceded by an argument.

^Z-C Suspend **tiler**.

^Z-D Delete PPL cell at current cursor location. With argument, deletes many cells, moving in the default direction. Does NOT delete breaks around the cell. To delete a cell and its breaks, use ^D.

^Z-F move Forward (right) one or more screens. May be preceded by an argument.

^Z-G (aborted command, rings bell only)

^Z-I (META Tab). Set MARK. Argument between 0 and 9. (Default 1) Saves the current cursor location for future reference.

NOTE: MARK 1 is the "default" mark, which may be changed by some of the window commands. For remembering "special" PPL coordinates for long periods of time, use any of the other MARKS. Use ^I or TAB to move to marks.

^Z-K Kill WINDOW. Argument between 0 and 9 (default 1). Delete the cells from the specified window, saving them on the kill ring. Blocks of cells which are on the kill ring may be retrieved with the ^Y (yank) and M-Y (rotate kill ring and yank again) commands. The ^K command requires the cursor to be at the window origin in order to execute the command, unless the ^K is preceded by a ^Q.

NOTE: This version does *not* erase breaks on the edge of the window. To delete the window and the breaks on the edges of the window, use ^K.

^Z-L Adjust screen. Moves the cell under the cursor to the upper-left corner of the screen.

^Z-N move down one or more screens. May be preceded by an argument.

^Z-O move to origin of PPL array, changing screens if necessary.

^Z-P move up one or more screens. May be preceded by an argument.

^Z-W Write WINDOW. Argument between 0 and 9 (default 1). Save the PPL cells from the specified window in a file. Cursor must be at the window origin in order to execute this command, unless the command is preceded by ^Q.

^Z-Z Suspend **tiler**. **Tiler** may be re-entered without losing anything.

<esc><esc>,
M-<esc>,
M-^[,

^Z-[(<esc><esc>) Escape to/from USER mode. This command temporarily toggles USER mode for the duration of a single command. This command may be used as a command prefix to access the "other" keyboard. When in USER mode, this prefix causes the standard **tiler** command to be take priority over any keyboard macro that may exist for the given command. When not in USER mode, this prefix causes the USER command (if one exists) to take priority over the standard **tiler** command. This prefix is displayed on the command line as '<*>'.

^Z-_,
^Z-/,
^Z-? Display the last error message encountered.

4.4 eXtended commands:

^X-A move to Start of PPL. Moves to the first PPL cell or break character of the entire PPL row, changing screens if necessary. If the row has no cell or break characters, moves to the far left edge of the PPL array.

^X-B select Buffer. If followed immediately by <return>, **tiler** connects to 'previous' buffer. This allows the user to quickly switch back and forth between to buffers, without specifying the names of the buffers each time. If followed by the name of a previously created buffer, **tiler** connects to that buffer. If given a name that is different than all current buffer names, **tiler** creates a new (empty) buffer and connects to that buffer. Use ^X^B to list all buffers and associated file names.

^X-C CIF checks. Checks the entire PPL array for cell placement errors, and creates a .ERR file listing all errors found. The M-C command may be used to find and correct design rule errors one at a time.

^X-D Define key. Assigns the current keyboard macro to the specified key. *See ^X-E for more information about keyboard macros.*

^X-E Execute keyboard macro. May be preceded by an argument to repeat the macro execution. This feature allows a special command to be defined by the user, and used just like any other **tiler** command. The keyboard macro is a stored sequence of commands typed by the user. A keyboard macro definition is started by the ^X-(command. Commands used after that time are executed in the normal fashion, but are also stored for future use as part of the keyboard macro. The definition of the keyboard macro is completed by the ^X-) command. After a complete keyboard macro has been defined, it may be executed by the ^X-E command.

^X-F list File. Displays the contents of a text file on the screen. This is

useful for examining files without exiting **tiler**. Especially useful for looking at .err error files created by the ^X-C (CIF check) command or the ^X-G (generate CIF) command, but may be used with any text file.

^X-H Mark whole buffer. Selects the entire PPL array as the default WINDOW.

^X-I Insert PPL file. **Tiler** prompts for a file to be inserted at the current cursor location in the PPL array. Blocks developed in previous editing sessions can be inserted into a PPL design using this command.

^X-K Describe User Key. **Tiler** prompts for the USER command and displays the keyboard macro currently defined for that command. If no USER defined key sequence has been assigned to that key, then "(Key not defined.)" is displayed. Note that the key sequence stored by **tiler** may not be identical to the key sequence used to create the keyboard macro, since **tiler** edits the key sequence as it is created, to remove redundant keystrokes.

^X-L show Location in PPL buffer. Displays the current cursor location (row, column) in the message area below the status line. Useful when coordinate echo is off.

^X-M Modify window. Sets the modifier value of all cells in the current current window to the value given as an argument.

^X-O Move to PPL Origin. Place cursor at row 0, column 0.

^X-T new Technology. This command tells **tiler** to read in a new TECHNOLOGY file and start a new editing session with the new technology.

^X-U Undefine key. Removes a key definition from the specified key. See ^X-D and ^X-E commands for more information on defined keys.

^X-X eXecute command file. Prompts for the name of a file containing **tiler** commands to be executed. An ordinary text editor may be used to create a **tiler** command file. The keystrokes stored in a command file are the same keystrokes used to execute **tiler** commands from the keyboard. Key definitions can be stored in command files so that commonly used keyboard macros may be loaded into **tiler** without requiring the user to manually type in the key definitions. (A future version of **tiler** will also include automatic execution of a 'tiler.ini' file whenever **tiler** is started, and allow logging of **tiler** commands typed from the keyboard, so that command files may be created from **tiler** as well)

^X-= show Location in PPL buffer. Displays the current cursor location (row, column) in the message area below the status line. Useful when coordinate echo is off.

^X-(Begin defining keyboard macro. After you type this command, every command you type is both executed and saved in the keyboard buffer, as part of the definition of the keyboard macro. The definition of the keyboard macro is completed by the ^X-) command. A completed keyboard macro may be executed by the ^X-E command. When the ^X-E command is typed, the commands saved in the keyboard buffer are executed just as if they had been typed in by the user. The keyboard macro may be used as often as desired, and is not altered until it is replaced by a new keyboard macro. The ^X-E command may be preceded by an argument, which will cause the keyboard macro to be repeated.

^X-) End of keyboard macro. This command completes the keyboard macro that was started with the ^X-(command. The commands typed between the ^X-(and the ^X-) can now be executed by typing ^X-E.

^X-[Move to the origin (lower-left corner) of the PPL array, and select the entire PPL array as the default window.

^X-] Move to the upper-right corner of the PPL array, and select the entire PPL array as the default window.

^X-{ Set lower-left corner of PPL array. Reduces the size of the PPL array so that he current cursor position becomes the lower-left corner of the PPL array. **USE WITH CAUTION!** All PPL cells below the cursor and left of the cursor are permanently removed from the PPL array.

^X-} Set upper-right corner of PPL array. Reduces the size of the PPL array so that the current cursor position becomes the upper-right corner of the PPL array. **USE WITH CAUTION!** All PPL cells above the cursor and right of the cursor are permanently removed from the PPL array.

^X-/,
^X-? HELP! Invokes the help utility, which can give general information about **tiler**, list all **tiler** commands, list available cells, describe specific commands or cells, and perform other fascinating informational tasks.

^X^A move Backwards to the left edge of the PPL array.

^X^B list Buffers. Lists the names of all buffers and the names of the files in each buffer. A '*' next to the file name indicates that the PPL array in that buffer has been modified, but the changes have not been saved on disk.

^X^C exit **tiler** entirely. If changes have been made in any buffers, and those changes have not been saved in files, the user will be warned that "Modified buffers exist. Exit anyway?" and **tiler** will wait for a response

from the user. Typing the 'y' key will cause **tiler** to exit WITHOUT saving the modified buffers. Typing any other key (such as 'n') will cause the ^X^C command to be ignored, and continue the editing session so that the user can save modified buffers (with ^X^S or ^X^W) before exiting.

^X^E move to End of PPL. Moves past the last PPL cell or break character of the entire PPL row, changing screens if necessary.

^X^F Find file. Find the specified file, and edit that file in its own buffer. If that file already exists in a named buffer, **tiler** simply connects to that buffer. Otherwise the file is read into its own buffer, and **tiler** connects to that buffer. Use ^X-B to select a different buffer, and ^X^B to list the buffers with the associated file names.

^X^G (aborted command, rings bell only)

^X^I Insert PPL rows or columns. Inserts entire rows or columns into the PPL array, splitting the PPL apart. This increases the size of the PPL array. The number of rows/columns to be inserted may be passed to the command as an argument. This command will prompt for an "insert direction". This direction specifies on which side of the cursor the new rows/columns are to be inserted. Use ^P or ^N when new rows should be inserted above or below the "current" row, and use ^B or ^F when new columns should be inserted to the left or right of the "current" column.

^X^K Delete PPL rows or columns. Removes entire rows or columns from the PPL array, moving the remaining PPL blocks together. This reduces the size of the PPL array, and should be used carefully. The number of rows/columns to be deleted may be passed to the command as an argument. This command will prompt for a "delete direction". This direction specifies on which direction specifies on which side of the cursor the deletion is to take place. Use ^P or ^N when rows should be deleted above or below the "current" row, and use ^B or ^F when columns should be deleted to the left or to the right of the "current" column. **USE WITH CAUTION!** All PPL cells deleted rows/columns are permanently removed from the PPL array.

^X^L Log **tiler** commands. When given with no explicit argument or with arg > 0, this command prompts for the name of a file for storing each KEYSTROKE typed into **tiler**. The log file created by this command can be used to recover from major blunders or system crashes. When given an arg < 0, logging is suspended temporarily. Logging may be resumed by giving ^X^L again with no argument, or arg > 0. An argument of 0 stops all logging and closes the log file. Log files may be "played back" with the ^X-X command.

^X^O move to PPL origin (row 0, column 0).

^X^R Read PPL file. This starts a new edit session with the specified file. If the PPL array in the current buffer has been modified, **tiler** offers to save the results of the previous editing session before reading the new array into the buffer. To edit a file in a different buffer, use ^X^F or ^X^V. **Tiler** can work with several files/buffers simultaneously.

^X^S Save changes in the current file. This commands writes out the PPL array in the connected buffer to the file indicated in the STATUS line, and returns for more editing of the same file. If the connected buffer has no file name associated with it, an error message is displayed. Use ^X^W to write specify a file name for writing out the connected buffer.

^X^V Visit file in its own buffer. Same as ^X^F.

^X^W Write file. **Tiler** prompts for the file to be used for storing the contents of the connected buffer. This file name then becomes the default file associated with the buffer. The file may be updated periodically with the ^X^S command.

^X^X Exchange POINT and MARK. Argument between 0 and 9 (Default 1). Exchanges the current cursor position and the specified MARK.

^X^Z Suspend **tiler**. **Tiler** may be re-entered without losing anything.